## Source: Intel Corporation, Huawei Technologies Co., Ltd (Rapporteur)

## Title: [MP\_RTT] Proposed Permanent Document Update v0.2.0

## Version: 0.2.0

## Agenda Item: 14.11

## Document for: Discussion and Agreement

Table of Contents

[1. Introduction 2](#_Toc132964874)

[2. Introduction to Multi-party RTT 2](#_Toc132964875)

[2.1 Multi-party RTT use cases and scenarios 2](#_Toc132964876)

[2.2 Multi-party RTT requirements 3](#_Toc132964877)

[3. Multi-party RTT Solutions 4](#_Toc132964878)

[3.1 Multi-party RTT over RTP Solution 4](#_Toc132964879)

[3.2 Multi-party RTT over IMS Data Channel Solution 4](#_Toc132964880)

[4. Comparison between RTP and IMS Data Channel Solution 15](#_Toc132964881)

[5. Interworking for Multiparty RTT between RTP and IMS Data Channel Solution 15](#_Toc132964882)

[6. KPIs 15](#_Toc132964883)

[7. References 16](#_Toc132964884)

# Introduction

The proposed use cases and requirements proposed in [SA4R230052](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR230052.zip) were agreed during RTC SWG post 122 #9 telco and the proposed permanent document update in [S4aR230061](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR230061.zip) was agreed upon during RTC SWG post 122#10 telco. In SA4#12e-3, [S4-230704](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_123-e/Docs/S4-230704.zip) is also agreed.

This version (v0.2.0) implemented the above agreed Tdocs.

The objective of this work item is to specify suitable solutions for multiparty RTT media in IMS, for both RTP and IMS data channel transport.

The concrete objectives are as follows:

* Collect and document detailed use cases for multiparty usage of RTT
* Develop harmonized solutions for both RTP and IMS data channel transport that address the detailed use case needs
* Amend existing IMS control/signalling flows to support the solutions, if found necessary
* Document pros and cons of each solution, and provide implementation guidelines to equipment vendors, as an informational Annex
* Inform/coordinate with at least SA2, CT1, CT4, and with other relevant 3GPP groups as found necessary, to enable alignment and possible updates of specifications under the responsibility of those groups.

# Introduction to Multi-party RTT

### 2.1 Multi-party RTT use cases and scenarios

According to clause 6.4 of [Draft - DTR/HF-00103708 v0.0.11](https://docbox.etsi.org/HF/HF/05-CONTRIBUTIONS/2022/HF%2822%29088017_Draft_-_DTR_HF-00103708_v0_0_11_TR_103_708.zip) [1], Multiparty RTT use cases

are defined as follows:

* Call using RTT within a small group of Deaf persons
* Deaf person calling emergency service and using RTT
* Hard-of-hearing user talking with hearing friends
* Deaf user participating in conference getting transcription support
* Deaf user participating in conference contributing by text-to-speech
* Deaf-Blind user participating in remote meeting
* Person in a critical situation making an emergency call by RTT
* Person in remote group meeting in occasional noise
* Relay service using multiparty technology
* Using an RTT relay service to connect to a voice conference call

### 2.2 Multi-party RTT requirements

According to clause 3 of [draft-hellstrom-avtcore-Multiparty-rtt-solutions-08](https://www.ietf.org/archive/id/draft-hellstrom-avtcore-multi-party-rtt-solutions-08.html#name-centralized-conference-mode)[2], the key Multiparty RTT requirements are listed as follows:

**General requirements:**

* A solution shall be applicable to IMS, SIP-based VoIP, and Next Generation Emergency Services.
* The transmission interval for text should not be longer than 500 milliseconds when there is anything available to send. Ref ITU-T T.140.
* If text loss is detected or suspected, a missing text marker should be inserted in the text stream.
* The display of text from the members of the conversation shall be arranged so that the text from each participant is clearly readable, and its source and the relative timing of entered text is visualized in the display. Mechanisms for looking back in the contents from the current session should be provided. The text should be displayed as soon as it is received.
* Bridges must be multimedia capable (voice, video, text).
* It MUST be possible to use real-time text in conferences both as a medium of discussion between individual participants (for example, for sidebar discussions in real-time text while listening to the main conference audio) and for central support of the conference with real-time text interpretation of speech.
* It should be possible to protect RTT contents with the usual means for privacy and integrity.

**Performance requirements:**

* The mixer performance requirements can be expressed in one number, extracted from the user requirements on real-time text expressed in ITU-T F.700, where it is stated that for "good" usability, text characters should not be delayed more than 1 second from creation to presentation. For "usable" usability the figure is 2 seconds.
* The mean delay of text passing the mixer introduced when only one participant is sending text should be kept to a minimum and should not be more than 400 ms.
* The mean delay of text passing the mixer should not be more than 1 second during moments when up to three users are sending text simultaneously.
* For the very rare case that more than three participants send text simultaneously, the mixer may take action to limit the introduced delay of the text passing the mixer to 7 seconds.
* The load on network and nodes should be limited. This is usually achieved by setting a limit for how many packets per second that may be sent from a mixer to each participant. While two-party use by RFC 4103, limits the load to 3.3 packets per second, a realistic limit for mixers could be 10 packets per second.

# Multi-party RTT Solutions

### 3.1 Multi-party RTT over RTP Solution

### 3.2 Multi-party RTT over IMS Data Channel Solution

#### 3.2.1 Architecture



Figure 3.2.1-1 Multi-party RTT over IMS Data Channel Architecture

The Multi-party RTT over data channel solution is based on data channel architecture, which is defined in clause AC.2.1 of TS 23.228 [3].

According to clause 5.5 of RFC8865 [4], for multiparty considerations, two alternatives were considered when searching for an efficient and easily implemented multiparty method for real-time text:

Multiple DC streams, one per participant:

One DC stream per source would be sent in the same session. UE can identify the source by the “label” attribute in the DC stream ID line when receiving RTT. If a new UE is added to the conference, a new downlink stream ID indicating the new UE should be added to all the existing participants. The conference application needs to manage the mapping relationship between the UE identity and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant.

**Pros:**

This is a straightforward solution. The load per source is low.

**Cons:**

With a high number of participants, the overhead of establishing and maintaining the high number of data channels required may be high, even if the load per channel is low.

Single DC stream, each participate use only one DC stream:

Only one DC stream for each participate, no SIP negotiation procedure for each participant when a new UE is added to the conference. The conference server should add a source information in front of the RTT content by identifying the label attribute in the DC stream ID line when receiving RTT from a UE.

**Pros:**

No negotiation when a new UE is added to the conference.

**Cons:**

The conference server should add decode and re-encode the RTT content.

#### 3.2.2 Call Flow

##### 3.2.2.1 Multi DC Streams

An example for three participants in a conference:



Figure 3.2.2.1-1 Multi DC Streams Example

Each UE has one uplink stream ID and two downlink stream IDs, if a new UE is added to the conference, a new downlink stream ID indicating the new UE should be added to all the existing participants.

The conference application needs to manage the mapping relationship between the UE name and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant, and add the UE identity before the real-time text to correctly display the source.

##### 3.2.2.1.1 UE Aware Mode



Figure 3.2.2.1.1-1 Multi DC Streams with UE Aware Mode Call Flow

The steps are shown as below:

Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including 3 DC stream IDs, one ‘sendonly’ for UE-A sending RTT to other participants, one ‘recvonly’ for receiving UE-B’s RTT, and the last one ‘recvonly’ for receiving UE-C’s RTT, the label attribute in each ‘a=dcmap’ can be get from the conference information, which can identify each DC stream belongs to whom. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendonly

a=dcsa:200 hlang-send:es eo

a=dcmap:201 label="B-Identity";subprotocol="t140"

a=dcsa:201 fmtp:t140 cps=20 recvonly

a=dcsa:201 hlang-recv:es eo

a=dcmap:202 label="C-Identity";subprotocol="t140"

a=dcsa:202 fmtp:t140 cps=20 recvonly

a=dcsa:202 hlang-recv:es eo

3. DCSF establishes corresponding DC stream IDs for UE-A.

4-5. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step2.

6-7. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step2.

Case 2: UE-D call into the conference and run the RTT application.

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

10. IMS-A establishes the DC stream for UE-D.

11. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

12-14. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

15. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

##### 3.2.2.1.2 UE Unaware Mode



Figure 3.2.2.1.2-1 Multi DC Streams with UE Unaware Mode Call Flow

The steps are shown as below:

Case 1: UE-A creates a conference and joins UE-B and UE-C into the conference, then runs the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one uplink DC stream ID with ‘sendonly’ for UE-A sending RTT to other participants, the label attribute in ‘a=dcmap’ can be get from UE-A’s identity. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendonly

a=dcsa:200 hlang-send:es eo

3. DCSF establishes corresponding DC stream ID for UE-A.

4. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add another two new downlink DC streams for UE-A, and three DC streams including one uplink DC streams and two downlink DC streams for the other participants.

5-6. IMS-A sends an REINVITE message adding two downlink stream IDs to UE-A and establish corresponding DC stream IDs for UE-A. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 recvonly

a=dcsa:200 hlang-send:es eo

a=dcmap:201 label="B-Identity";subprotocol="t140"

a=dcsa:201 fmtp:t140 cps=20 sendonly

a=dcsa:201 hlang-recv:es eo

a=dcmap:202 label="C-Identity";subprotocol="t140"

a=dcsa:202 fmtp:t140 cps=20 sendonly

a=dcsa:202 hlang-recv:es eo

7-8. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step4.

9-10. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step4.

Case 2: UE-D calls into the conference and run the RTT application.

11. UE-D calls into the conference created by UE-A, and runs the RTT application.

12. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

13. IMS-A establishes the DC stream for UE-D.

14. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

15-17. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

18. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

#### 3.2.2.2 Single DC Stream

An example for three participants in a conference:



Figure 3.2.2.2-1 Single DC Stream Example

T140 protocol is too old to be extended to support adding the source label, so the conference server can add a source label getting from the “label” attribute of ‘a=dcmap’ line in front of the text content when receiving the real-time text from a UE, and the terminal can display it directly without modification.

 

Figure 3.2.2.2-2 Single DC Stream Call Flow

The steps are shown as below:

Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including only one DC stream ID, the SDP offer example:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendrecv

a=dcsa:200 hlang-send:es eo

3. DCSF establishes corresponding DC stream ID for UE-A.

4-5. IMS-A sends an REINVITE message with only one stream ID to UE-B and establishes corresponding DC stream ID for UE-B. The stream ID is similar to step2.

6-7. IMS-A sends an REINVITE message with only one stream ID to UE-C and establishes corresponding DC stream ID for UE-C. The stream ID is similar to step2.

Case 2: UE-D call into the conference and run the RTT application.

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending and receiving RTT.

10. IMS-A establishes the DC stream for UE-D.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will identify the source by the application data channel established is between UE-A and DCMF/MRF, and then add the UE-A’s identity as source to the RTT content. DCMF simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D directly display the RTT content.

# Comparison between RTP and IMS Data Channel Solution

# Interworking for Multiparty RTT between RTP and IMS Data Channel Solution

# KPIs

# References

------------------------------------------------------------------------------

[1] Draft - DTR/HF-00103708 v0.0.11: “Human Factors (HF);Real-Time Text (RTT) in Multiparty conference calling”

[2] draft-hellstrom-avtcore-Multiparty-rtt-solutions-08: “Real-time text solutions for multi-party sessions”

[3] TS 23.228: “IP Multimedia Subsystem (IMS)”

[4] RFC8865: “T.140 Real-Time Text Conversation over WebRTC Data Channels”